Trabalho 1

Introdução ao Processamento Digital de Imagem (MC920/MO443)
Professor: Hélio Pedrini

Vinicius Couto Espindola | RA: 188115

30 de Março de 2020

1 Introdução

O trabalho visa apresentar problemas e conceitos simples voltados ao processamento digital de imagens. Os seguintes problemas serão discutidos: redução de resolução de imagens, quantização de imagens e transformações na escala de cinza. A fim de reduzir o número de amostras de uma imagem, aplicou-se métodos para preservar informação da imagem ao relizar tal redução. Ainda voltado a compactação de imagens, quantização visa reduzir o número de níveis necessário para representar uma imagens, em contrapartida, invés de reduzir o número de amostras necessárias para representação das imagens, reduz-se o tamanhode cada célula. Por fim, exploraremos equações para transformar a escala de cinza de uma imagem.

2 Execução

`python3 assignment_1.py <function> <argument-list> <input-file>`

function: Pode ser definida como qualquer uma das três funções implementadas no programa *assignment_ 1.py*: *resolution*, *quantization* e *greyscaler*.
input-file: Endereço da imagem de entrada a ser processada.
argument-list: Deve conter todos os parametros das função chamada em *<function>* separados por vírgulas. Os argumentos de cada função, são:

Quantization: int

- **1** Número de níveis de cinza para o qual reduzir a imagem.

Resolution: int

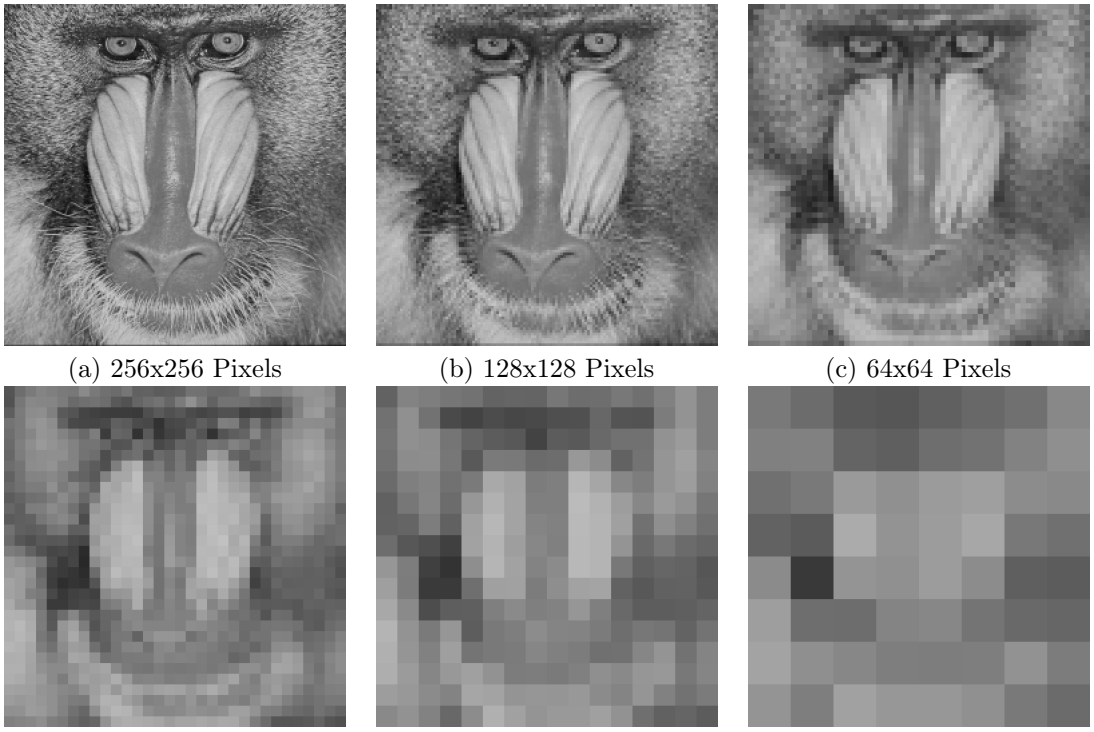
- **1** Número de amostras da dimensão para qual a imagem será reduzida.

Greyscaler: int,...

- **1** Indica qual função será utilizada na transformação.
- **Para** $int \in [0, 3]$**:** [0-3],<c-float>

  Para as funções de 0 à 3 apenas a opção e o valor da função C são passados como argumentos.
- **Para** $int = 4$**:** 4,<a-int>,<b-int>,$<\alpha - float>, <\beta - float>, <\gamma - float>$

  Para a função 4, deve-se definir todas as 5 constantes listadas.



(a) 256x256 Pixels (b) 128x128 Pixels (c) 64x64 Pixels

(d) 32x32 Pixels (d) 16x16 Pixels (d) 8x8 Pixels

Figure 1: Imagens com resulução reduzida

3 Soluções

3.1 Redução de Resolução

Dada uma imagem com com uma amostragem N linhas por N colunas, faremos a transformação desta para uma nova amostragem MxM, tal que M < N. O primeiro passo consiste em preservar informação usando uma quantia menor de dados. Suponha que queiramos reduzir a amostragem de uma imagem com $N = 16$ para $M = 4$, neste caso, teremos que representar blocos de 4x4 em apenas uma célula. Para esta redução de dados, optou-se por tirar a média do bloco de células e atribuir o resultado para uma única célula da nova amostragem. Formalizando o algoritimo, temos uma imagem I de amostragem NxN, reduziremos o número de amostras criando uma imagem I' com amostragem MxM. Para reduzir o número de amostras, agrupa-se as células em submatrizes disjuntas A_{ij} de dimensões KxK, tal que $K = N/M$ e calcula-se a média das células de cada submatriz.

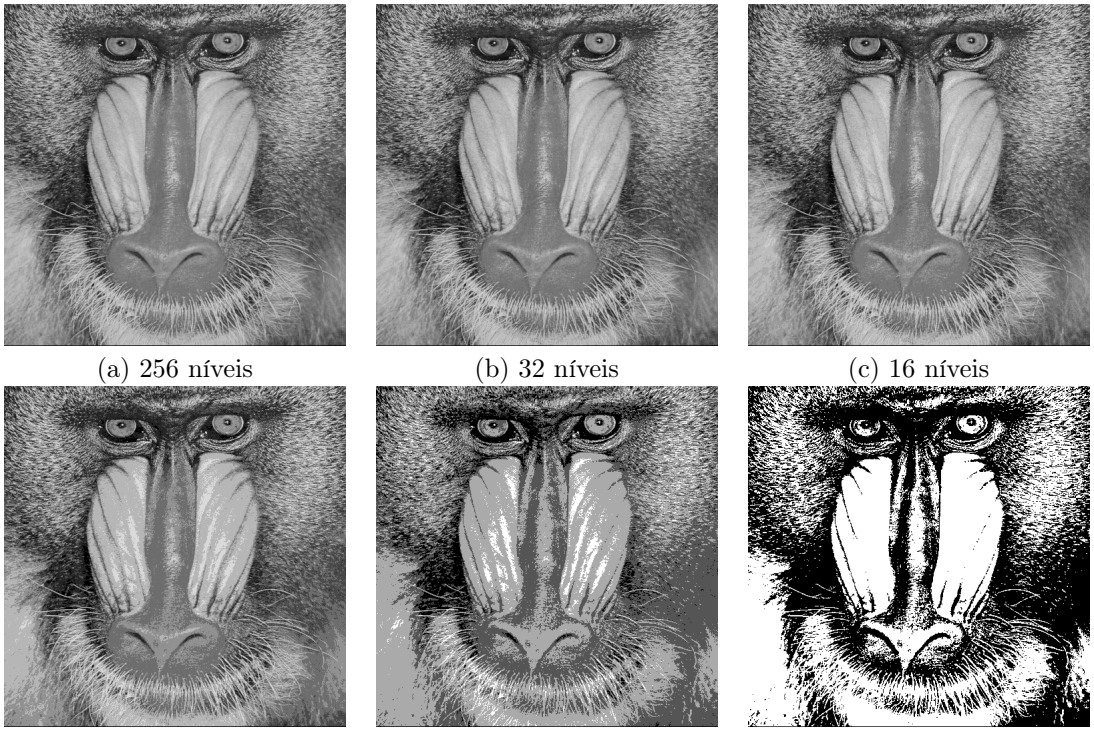
$$\forall i, j \in [0, M) \rightarrow I'_{ij} = media(A_{ij})$$

Para preservar as dimensões da imagen (área), fazemos: $A_{ij} = I'_{ij}$. Assim, atribuímos a média de cada submatriz A_{ij} a todas suas células, reduzindo a resolução da imagem I mas mantendo as dimensões originais. As os resultados obtidos utilizando este método estão ilustradas na **figura 1**, e podem ser reproduzidos com o seguinte commando:

```
python3 assignment_1.py resolution <dimension> ./baboon.png
```

3.2 Redução de Quantização

Uma imagem em que cada célula consiste de um Byte, possui 256 níveis de intensidade disponíveis, pois pode atribuir a cada célula um valor de 0 à 255. Diz-se, então, que a profundidade L de uma imagem, é a quantia de valores discretos representáveis por célula. O objetivo é reduzir e número de níveis L utilizados



(a) 256 níveis (b) 32 níveis (c) 16 níveis

(d) 8 níveis (e) 4 níveis (f) 2 níveis

Figure 2: Imagens geradas a partir da redução dos níveis de quantização

para representar uma imagem. Considerando que o valores são representados a partir de bits, temos que cada célula tem uma profundidade de $L = 2^k$, onde k é o número de bits por célula. Dada uma imagem I de profundidade L, uma profundidade L' tal que $L' < L$ e o número de bits por célula da imagem atual k, reduziremos o nível de quantização L da imagen de entrada para L'. Para aplicarmos tal redução, utilizamos a técnica de *binning* que envolve reduzir intervalos numéricos a um único valor. Supõe-se uma imagem com $L = 16$ e queremos redúzi-lá para $L' = 4$. Neste caso criaremos bins de tamanho L/L'=4 para agrupar os L níveis em L' . Temos, então:

$$range(0, L+1, L/L') = 0, 4, 8, 12, 16 \rightarrow B = \{b_0 = [0, 4), b_1 = [4, 8), b_2 = [8, 12), b_3 = [12, 16)\}$$

Para toda célula c(x,y) da imagem de entrada, e para cada b_x , aplica-se M(x,y)=x se $c(x, y) \in b_x$ Assim temos um mapa M que mapeia cada célula da imagem original para algum b_x, e como $|B| = 4$, podemos representar este mapa com apenas $L' = 4$ níveis de profundidade. Para representar L' em, por exemplo, imagens com $L = 256$, fazemos um mapeamento percentual de seus valores:

$$\forall_{x,y} M(x, y) = m, \text{ faça } \frac{m}{L' - 1} \cdot (L - 1) \text{ e arredonde para um inteiro}$$

Com isso, podemos "expandir" a imagem para uma profundidade L, tal que L>L', nota-se, no entanto, que há perda de informação nessa transformação $L \rightarrow L' \rightarrow L$. Os resultados obtidos com este médtodo estão ilustrados na **figura 2**. Estas imagens podem ser reproduzidas com o seguinte comando:

```
python3 assignment_1.py quantization <niveis> ./baboon.png
```

3.3 Transformação da Escala de Cinza

A seguir são definidas algumas funções que visam alterar a escala de cinza. Estas recebem o valor f do nível de cinza de um pixel e o transformam de acordo com a função especificada. As constantes das funções descritas foram ajustadas para que o valor máximo obtido fosse próximo à 255.

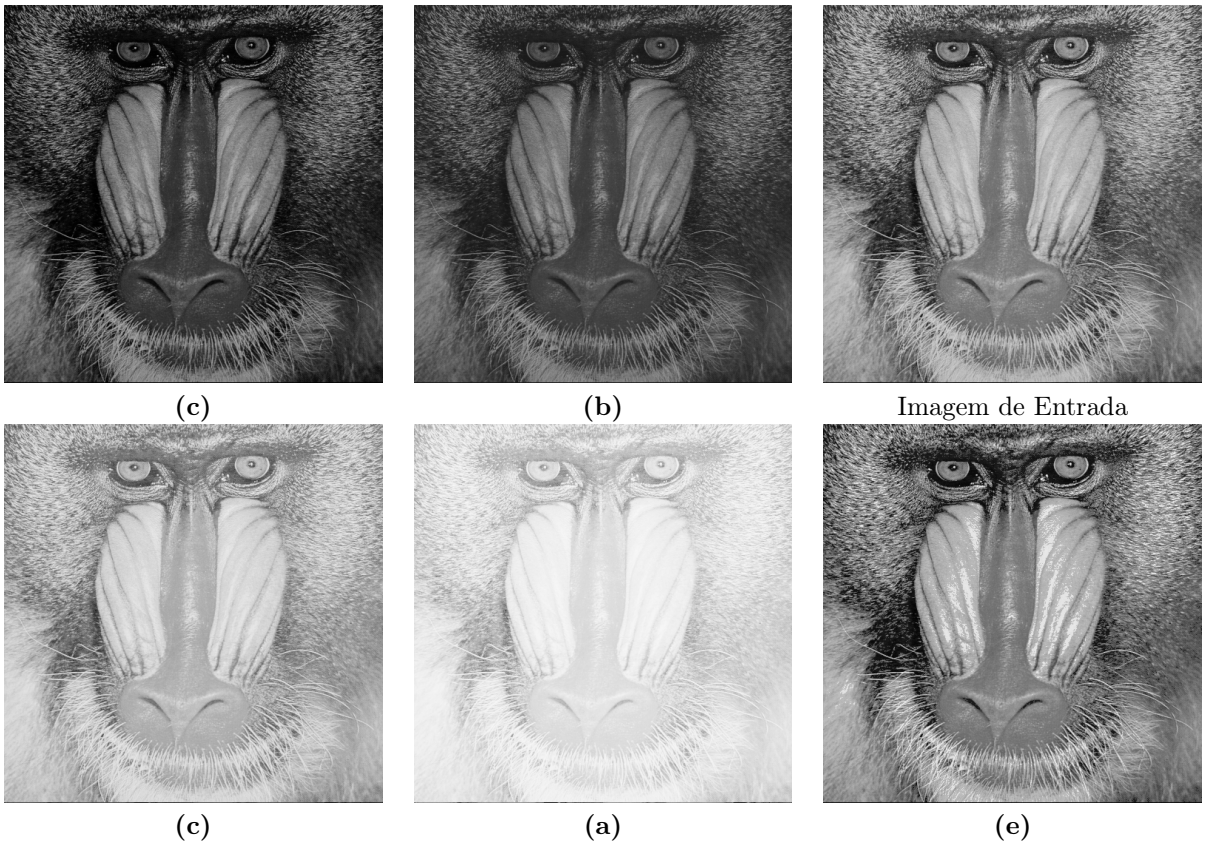


Figure 3: Imagens geradas a partir de tranformações nas escalas de cinza

- **(a)** $f_{log}(f) = c \cdot log_2(f+1)$, com $c = 32$
- **(b)** $f_{exp}(f) = c \cdot e^{f/100}$, com $c = 19.9$
- **(c)** $f_{sqr}(f) = c \cdot f^2$, com $c = 0.0039$
- **(d)** $f_{srt}(f) = c \cdot \sqrt{f}$, com $c = 0.0039$
- **(e)** $f_{spreader}(f) = \begin{cases} \alpha f & \text{se } 0 \leq f \leq a \\ \beta(f-\alpha) + \alpha a & \text{se } a < f \leq b \\ \gamma(f-b) + \beta(b-a) + \alpha a & \text{se } b < f \leq L \end{cases}$
  Com $a = 64, b = 192, \alpha = 0.5, \beta = 1.5$ e $\gamma = 0.5$

A fim de evitar *overflows*, o valor do pixel na função exponencial foi normalizado para o intervalo de 0 à 255. Nota se, também, que as constantes da função *spreader* foram selecionados de forma a obter um conjunto de retas continuas. As imagens produzidas a pratir das tranformações com constantes definindas se encontram a na **figura 3** indexadas pela letra conrrespondente à listagem das funções. Estas imagens podem ser reproduzidas com os seguintes comandos:

```
python3 assignment_1.py greyscaler <options-0to3>,<C> ./baboon.png
python3 assignment_1.py greyscaler 4,<a>,<b>,<alpha>,<beta>,<gamma> ./baboon.png
```

3.4 Limitações

Para o caso de redução de resolução, sempre assume-se que a imagem será reduzida há uma dimensão múltipla e menor que a original. Caso estes valores não sejam mútiplos a imagem terá apenas uma subcconjunto

de células com suas amostragens reduzidas. Nota-se também que o método implementado está limitado a imagens quadradas apenas. A função *resolution* também apresenta limitações quanto a perfomance: a iteração pelas submatrizes A_{ij} não está vetorizada. Não foi indetificada uma forma de vetorizar está operação. A redução de quantização também sempre assume que o número de níves para o qual a imagem será reduzida é menor do que o original. A príncipio, a função de quantização aceita apenas imagens com 8 bits por célula, mas isto pode ser alterado no parâmetro *bits* da função.
As transformações da escala de cinza não tratam casos de *overflow*. Caso sejam escolhidas constantes inadequadas para a trasnformação em questão, os pixels podem exeder o valor limite de 255 e serem convertidos para preto (que é o padrão para *overflow* no numpy). Diz-se então que as tranformações estão limidas à escolha adequada dos parâmetros destas.

4 Discussão e Conclusão

Releva-se a importância dos aspectos fundamentais de processamento digital de imagens e suas carácteristicas práticas. Quanto aos processos de amostragem e quantização, é bem clara a troca entre as carácteristicas de qualidade e aspectos de performance. Em amostragens densas temos que tratar um número de células consideravelmente maior. Quanto a quantização, nota-se que uma representação adequada da prefundidade de uma imagemdepende diretamente da quantização da imagem. Em contrapartida, pode-se otimizar diversos aspectos quanto a performance e memória observando, por exemplo, que a diferença entre imagens de 256 para 32 níveis de quantização não são tão claras ao olhar humano, e a segunda requer um oitavo do espaço necessário para ser armazenada em memória. No que disrespeito às transformações, nota-se como funções podem modificar de maneira controlada imagens as quais seria inviável alterar manualmente. Com isso, pode-se otimizar a qualidade da imagem tendo em vista um objetivo específico.

References

[1] R.C. Gonzalez. *Digital Image Processing*. Prentice Hall, 2007.